CPSC 468 Midterm Review

Chapter 1

Definition 0.1 (Turing Machine). *A k-tape Turing Machine M is described by a tuple (Γ, Q, δ). Assume $k \geq 2$, with 1 read-only input tape and $k-1$ work tapes. The last work tape is assumed to be the output tape.*

- Γ *the finite alphabet of symbols that M may have on its tapes. Assume that Γ contains at least $\{0, 1, \Box, \triangleright\}$.*
- Q *a finite set of possible states M's state register may be in. Assume that Q contains a q_{start} and q_{halt}.*
- $\delta : Q \times \Gamma^k \to Q \times \Gamma^{k-1} \times \{L, S, R\}^k$ *a transition function for M that takes in the current state and each head's read, and outputs the next state, with $k-1$ writes on all the work tapes, and movement direction for all k tapes.*

Definition 0.2 (Computing a Function). *Let $f : \{0,1\}^* \to \{0,1\}^*$ and $T : \mathbb{N} \to \mathbb{N}$, with M a TM. We say that M computes f if for every $x \in \{0,1\}^*$, if M is initialized to the start configuration on input x, then it halts with $f(x)$ on the output tape. We say M computes f in $T(n)$-time if its computation on every x requires at most $T(|x|)$ steps.*

Definition 0.3 (Time Constructible). *A function $T : \mathbb{N} \to \mathbb{N}$ is time constructible if $T(n) \geq n$ and there is a TM M that computes the function $x \mapsto \llcorner T(|x|) \lrcorner$ in time $T(n)$. $T(n) \geq n$ is to allow the algorithm to read its input.*

Example 0.1 (Time Constructible Functions). *Some time constructible functions are n, $n \log n$, n^2 and 2^n.*

Claim 0.1. *For every $f : \{0,1\}^* \to \{0,1\}$ and time constructible $T : \mathbb{N} \to \mathbb{N}$, if f is computable in time $T(n)$ by some TM M using alphabet Γ, then it is able to compute the same function using $\{0, 1, \Box, \triangleright\}$ in $(c \log_2 |\Gamma|) \cdot T(n)$. This is because we may express each symbol of Γ using $\log |\Gamma|$ binary bits, with some constant c overhead.*

Claim 0.2. *A k-tape TM can have its $k-1$ work tapes simuliated by a single tape by interleaving the k tapes together.*

Definition 0.4 (Oblivious Turing Machine). *An oblivious TM's head movement depends on the length of the input, not the contents of the input. Every TM can be simulated by an oblivious TM.*

Claim 0.3 (Turing Machines as Strings). *Every binary string $x \in \{0,1\}^*$ represents some TM, and every TM is represented by infinite such strings (think: comments in a language). The machine represented by x is denoted M_x.*

Claim 0.4 (Universal Turing Machine). *There exists a TM $\mathcal{U}$ such that for every $x, \alpha \in \{0,1\}^*$, $\mathcal{U}(x, a) = M_\alpha(x)$, where M_α denotes the TM represented by α. Moreover, if M_α halts on input x within T steps, then $\mathcal{U}_\alpha(x)$ halts within $CT \log T$ steps, where C is a number independent of $|x|$, and depends only on M_α's alphabet size, number of tapes, and number of states. In other words, the cost of simulating any machine M_α has a logarithmic overhead.*

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6